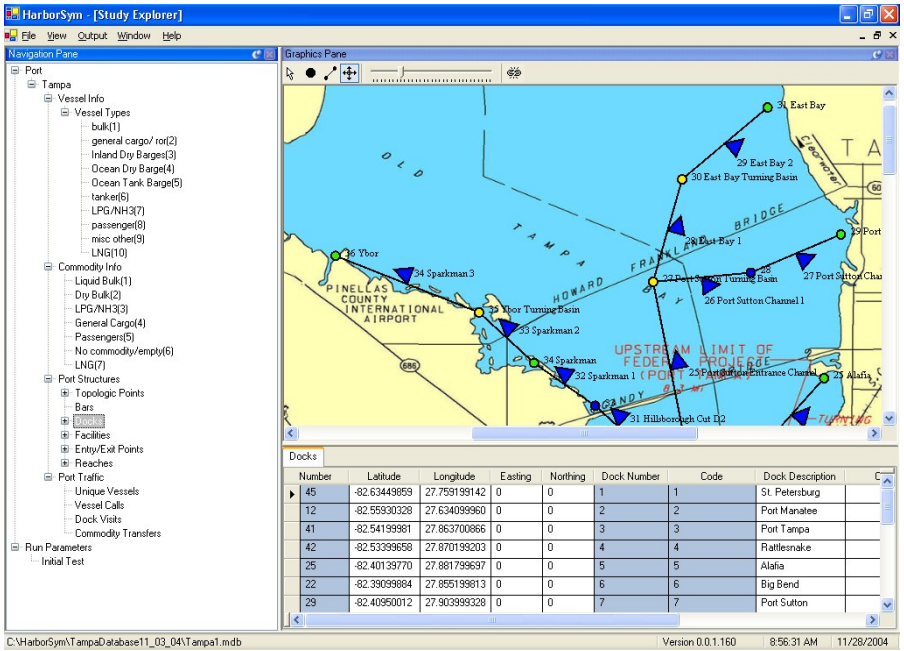
HSAM: An Interactive, Immersive Animation of Deep-Draft Maritime Traffic Simulations

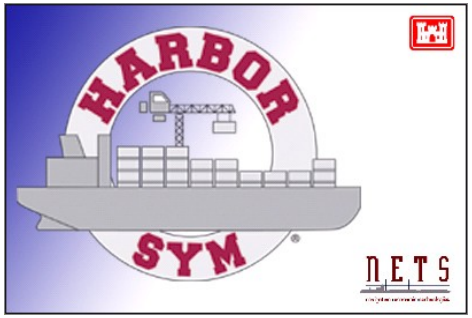
Keith Hofseth
Shana A. Heisey-Olig
Cory M . Rogers
William K. Woelbeling
Richard M. Males

HarborSym Basics

- U.S. Army Corps of Engineers (USACE) model for investigating feasibility of improvements in a port
- Monte-Carlo simulation of vessel movements within a port

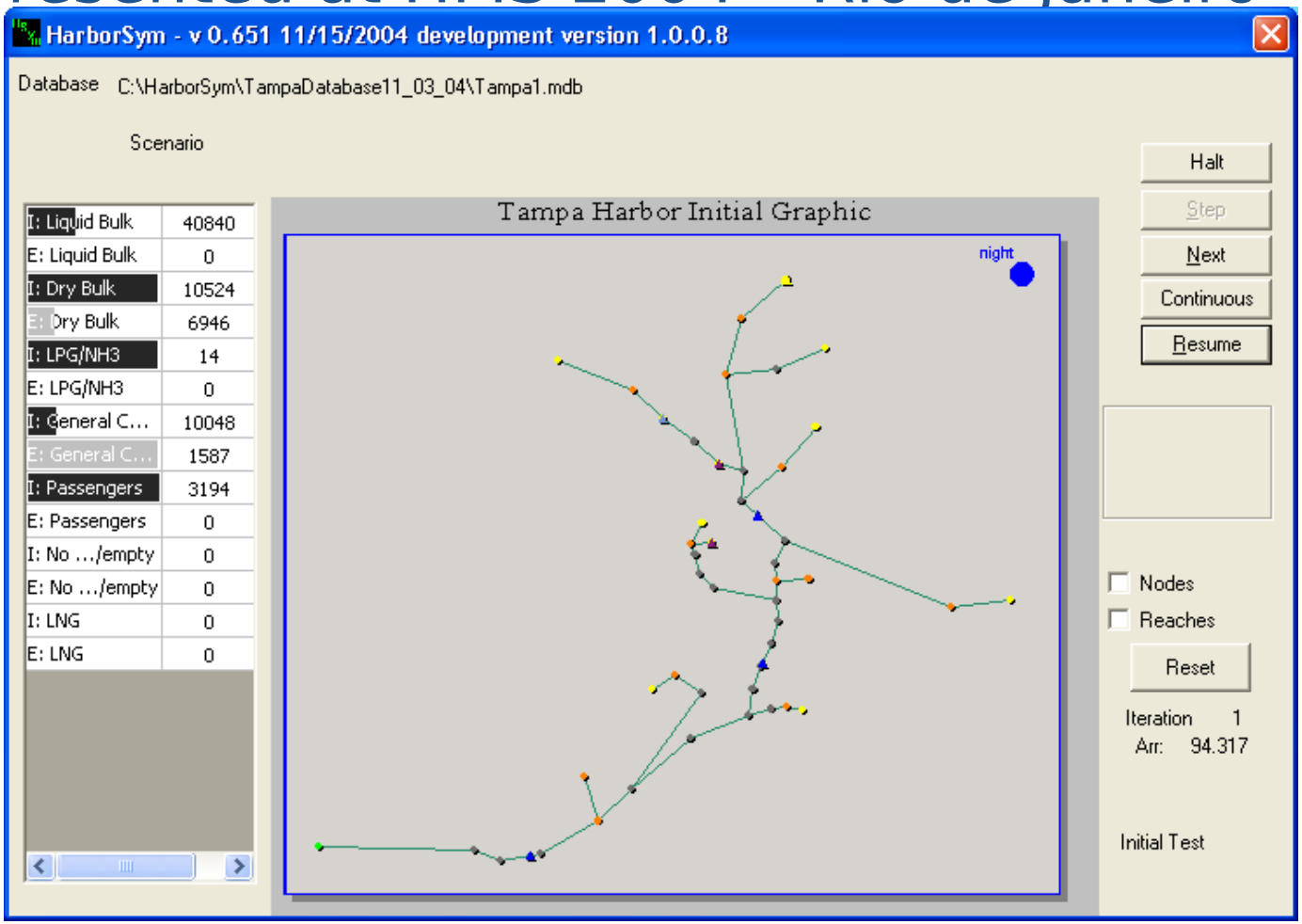


- Benefits captured through reductions in transit times



HarborSym Simulation

- Presented at HMS 2004 – Rio de Janeiro



Our Problem

- Competing Models Look Fantastic
- Embarrassment factor - Everyone laughing
- Development team resistant to “glitz”
  - Prefer function over form
- Large quantity of data
- Complex interactions between elements
- Model too intensive to provide quality visualization during simulation
- Our models need to be non-proprietary

The Mission

- Construct a dynamic, interactive, immersive 3-D environment based on HarborSym study data
  - At relatively low cost
  - In a freely distributable manner
  - No cost to end-user (no software to purchase)
  - Data driven to ensure portability without re-coding
  - Impress non-technical authoritative audience

HSAM Framework

- Post-Processing visualization of HarborSym simulation
- Move vessels through port as dictated by HarborSym simulation outputs
- Custom C++ application
- Open-Source Animation Engine
  - OGRE – Object-oriented Graphics Rendering Engine
    - Extensible
    - Freely distributable
    - Open source code (GNU license)
    - C++ / Fast
    - OpenGL and DirectX 9 supported for best utilization of advanced graphics cards

World Building

- Navigation Network
  - Network Configuration File
- The Fleet
- The Ground Plane
- Creating the 3D Illusion

Navigation Network

- Collection of points and reaches to produce a linked-node network
- Vessels move along reaches to traverse port
- Network configuration file
  - Defines the Navigation Network for a specific HarborSym study
  - Text file extracted from the HarborSym database

Network Configuration File

- Navigation Point Classes
  - Points have an ID from HarborSym
  - Points are named
  - Points have a type (way-point, turning area, dock)
  - Points have a location (X,Y,Z)
  - Points have unique avatars (3D representations)
    - Entry/Exit
    - Way-Point
    - Dock
    - Turning/Holding Area

Processing a POINT Command

- HarborSym provides a point (node) ID and coordinates of that point
- Avatar of correct Navigation Point Class is created in “The World” at the specified coordinates

- The Navigation Point is cataloged for referencing later in the simulation

Network Configuration File II

- Navigation Reaches
  - Reaches have an ID from HarborSym
  - Reaches are named
  - Reaches are linear between two Navigation Points
  - Reaches have direction (upstream/downstream)
  - Reaches have unique avatars

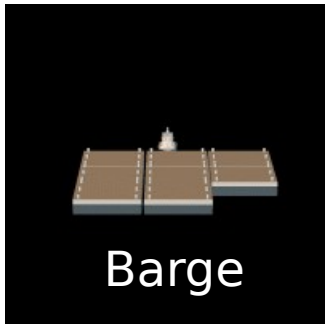
Processing a REACH Command

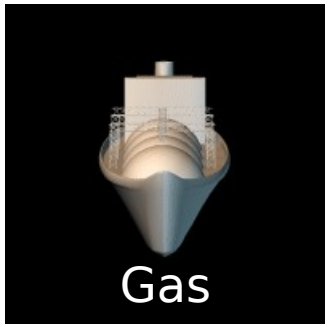
- HarborSym provides a Reach ID and two Navigation Points to define bounds

Avatar is created in “The World”

- Stretched to the proper length
- Rotated to “Connect the dots”

Fleet Construction





- Vessels have an ID from HarborSym
- Vessels are named
- Vessels have unique avatars by class
  - Location in 3-space (to start)
  - Orientation (upright)
  - Appearance (material applied to avatar)

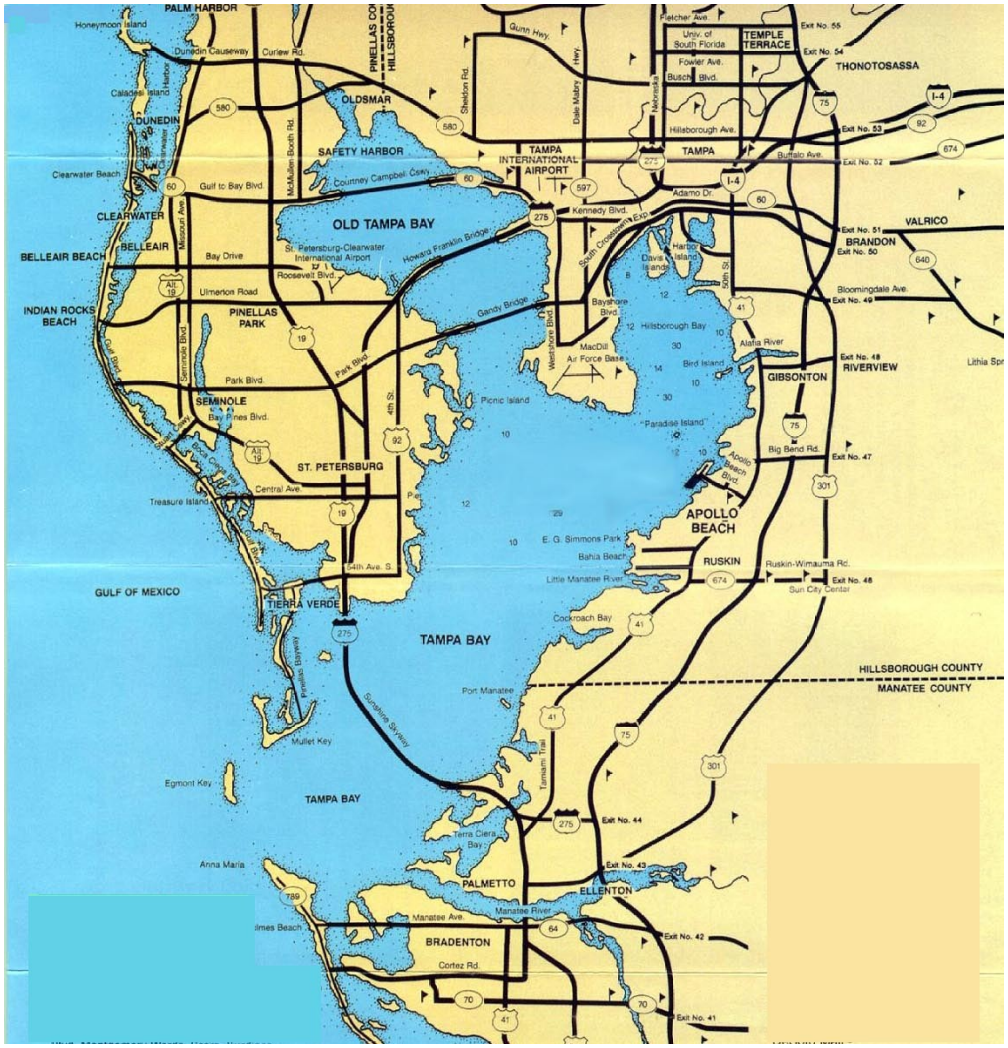
Fleet Construction II

- Vessels have a queue of movement commands
(a ToDo list)

- Reference Vessel Register
  - Created to describe the fleet used in the simulation
  - ID, Vessel Name, Vessel Class, Flag, Cargo List

The Ground Plane

- Surface over which the simulation plays out
- 20K x 20K square with a detailed image embossed on the surface
  - Small movements = smooth viewing
- Suspended in the center of the simulated world
- Overlaid with the Navigation Network



Creating the 3D Illusion

- Universe in a cardboard box
- Textures for sky, horizon, ground
- Lights on
- Camera ready
- Action
  - Electronic Flip-Book

The Rendering Loop

- Animation engine provides an event on the beginning and ending of the rendering of a frame
- HSAM provides an event handler for the event that is raised on the start of a frame being rendered
- For each frame
  - Read command from file
  - Place command in appropriate queue
  - Process vessel command queues
  - Process environmental command queue
- Next frame

Inside The Rendering Loop

- The Command List
  - Chronologically ordered list of directives
    - VTR, SUP, etc.
  - Affect either vessels in the simulation or the entire simulated world
  - Command file is a fixed width text file describing a HarborSym simulation run
  - Commands have start times and optionally end times
    - End time is optional for non-vessel movement commands such as “Sun Up”

Commanding “The World”

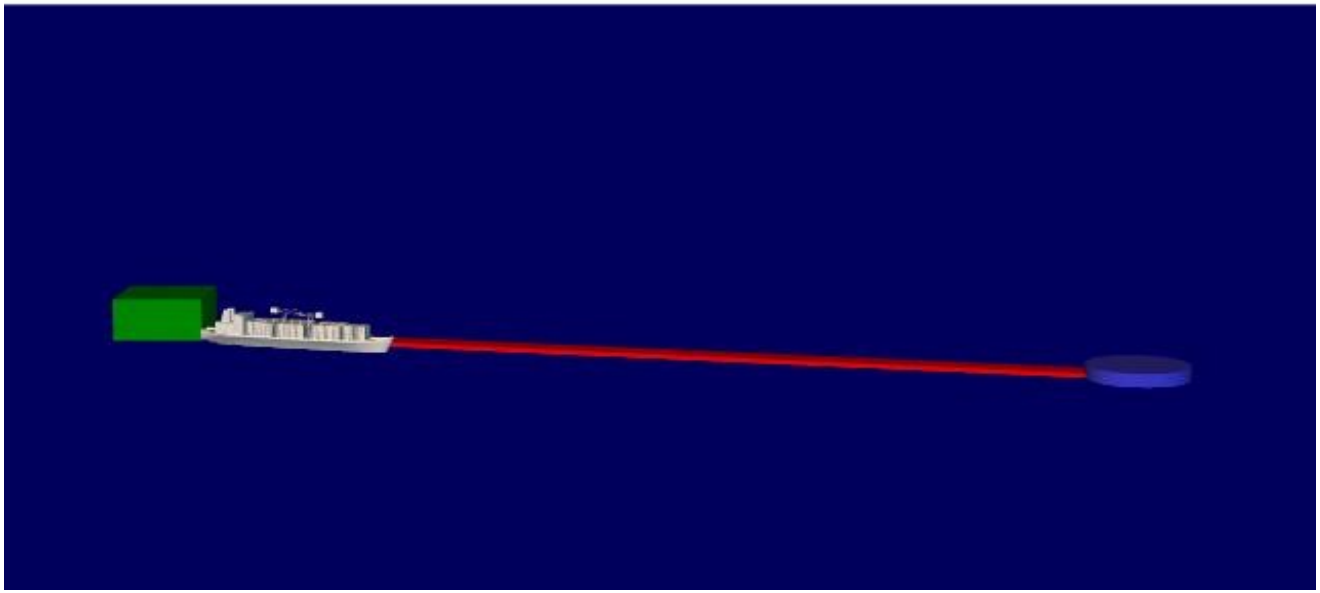
- Vessel Commands
  - Instructions for a vessel to move from point A to point B departing at one time and arriving at another
  - Start time, End time, Start point, End point
  - Vessel status during movement
    - “Under way”
    - “Cargo Transfer”
    - “Delayed by Traffic,” etc.
  - Destroy-on-completion indicator (end of voyage)
    - Distinguish between end of voyage and long-term parking

Processing Vessel Command Queues

- Each vessel in the simulation has a queue of movement commands
- Vessel queues are polled at each clock tick and processed accordingly
- Commands compared to The World clock to see if the command is still in-progress (if not, it is discarded)
- The vessel’s position is determined by the ratio of time into the command to current time and the distance between the start and end points of the command

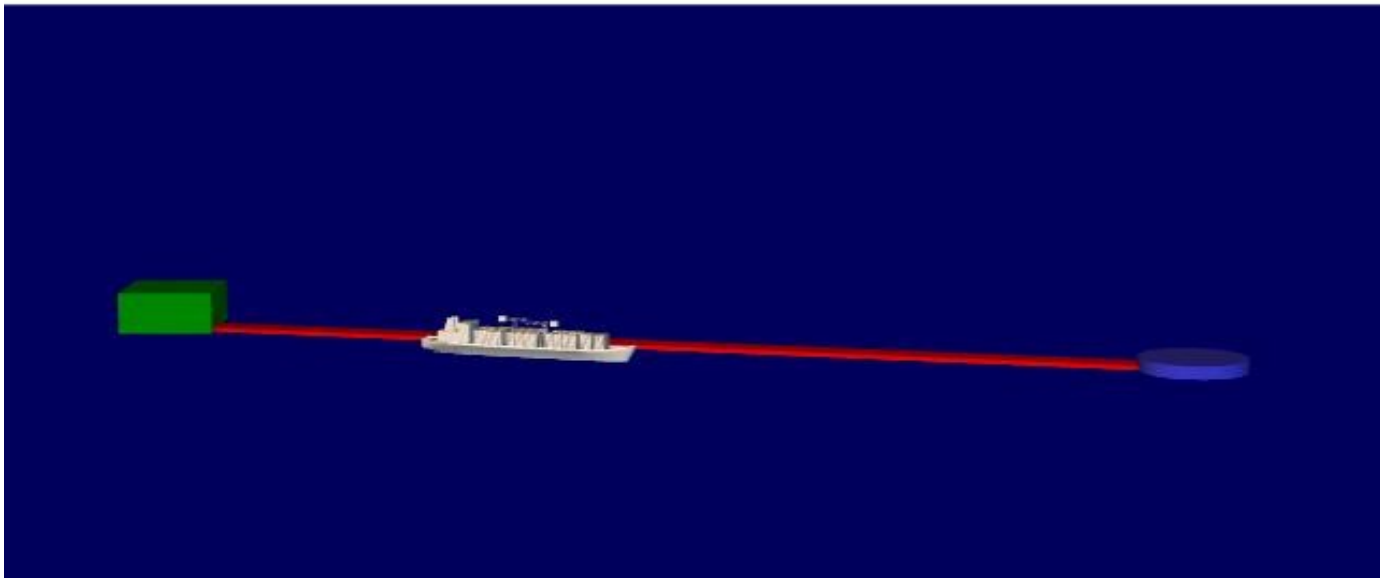
Vessel Movement I

- Vessel Begins Movement



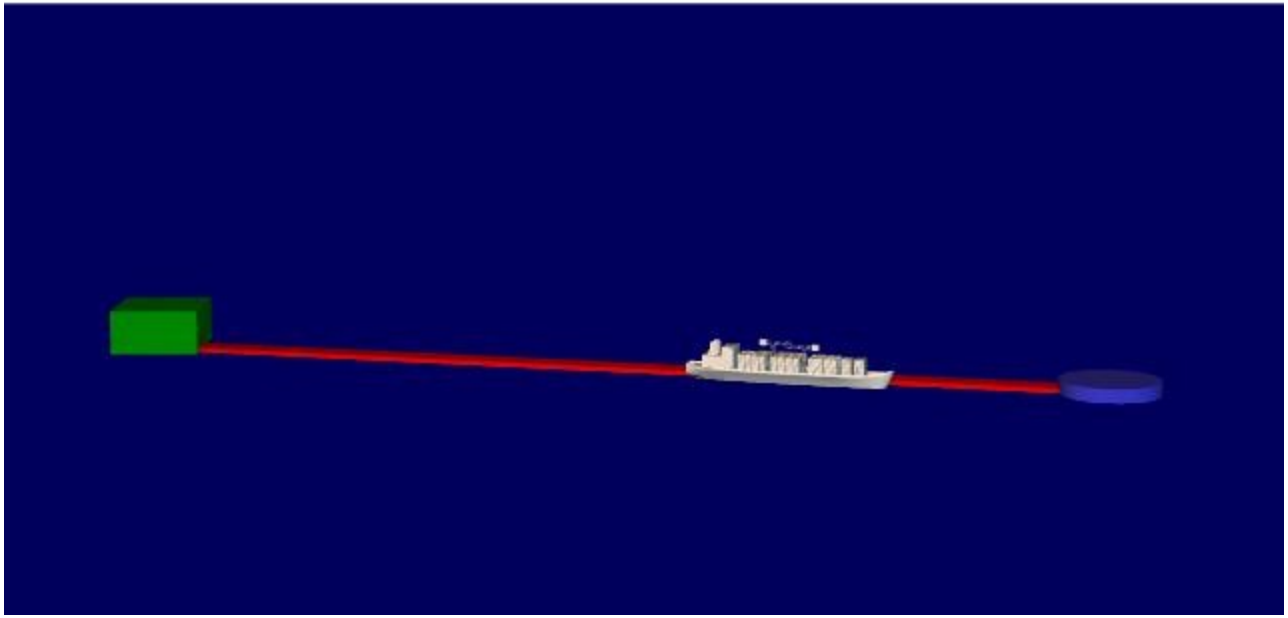
Vessel Movement II

- 25% of transit time elapsed – 25% of distance is traveled



Vessel Movement III

- 75% of transit time elapsed - 75% of distance is traveled



Commanding “The World” (continued)

- Environment Commands
  - Instructions to alter the appearance of simulated world without respect to a vessel
  - Command Code and Trigger Time
    - “Raise Sun” 5:30
    - “Lower Sun” 20:30
    - “Tide High” 6:25
    - “Tide Low” 13:15
    - Etc.

Processing Environmental Commands

- The Environment has a command queue
- The trigger time of the command at the top of the list is compared with current world time
  - world time < trigger time – do nothing
  - world time > trigger time – process command and remove from the command queue

Processing User Interaction

- Initial configuration file sets the appearance
- User controls
  - Time flow (faster/slower)
  - Camera position (flight controls)
  - Pause and interrogate vessels

Setting it All in Motion

- Construct initial world
- Start the World Clock
- Render Frames (~200 fps today)
  - Process simulation command and distribute
  - Process vessel queues
  - Process environment queue
  - Compare commands in queues to World Clock and move objects if necessary (vessels, sun, etc.)
  - Check for user commands (keyboard, mouse, etc..)

Concluding Remarks

- HSAM is a valuable tool for
  - Extraction of knowledge from complex system interactions
  - Identification of data anomalies, simulation problems, inconsistencies, etc…
  - Presentation and engagement of non-technical decision makers
- Data-driven architecture can be extended to other problems with minimal technical expertise in computer generated animation
- More information on OGRE can be obtained at
  - http://www.ogre3d.org

SYSTEM DEMONSTRATION

HSAM: An Interactive, Immersive Animation of Deep-Draft Maritime Traffic Simulations

Keith Hofseth
Shana A. Heisey-Olig
Cory M . Rogers
William K. Woelbeling
Richard M. Males